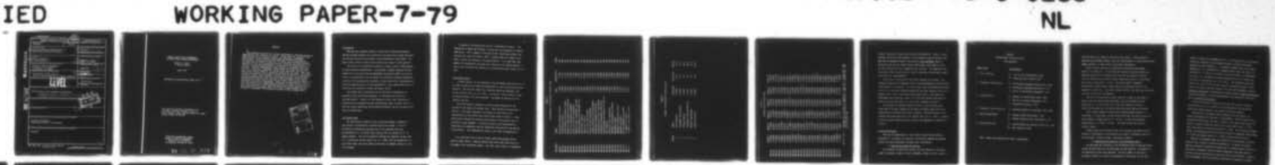
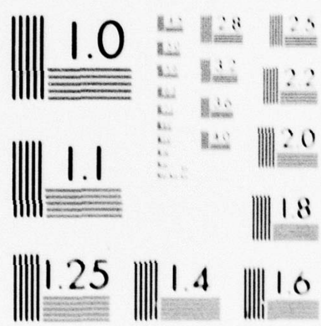
AD-A068 968 CALIFORNIA UNIV LOS ANGELES GRADUATE SCHOOL OF MANAGEMENT F/G 9/2
PROBLEM FACTORS AND DETERMINANTS IN APPLICATION SOFTWARE MAINTE--ETC(U)
MAR 79 B P LIENTZ, E B SWANSON N00014-75-C-0266
UNCLASSIFIED WORKING PAPER-7-79 NL

| OF |

AD
A068968



END
DATE
FILMED
7-79
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

12

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Problem Factors and Determinants in Application Software Maintenance.		5. TYPE OF REPORT & PERIOD COVERED (9) Technical rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Bennet P. Lientz, E. Burton Swanson		8. CONTRACT OR GRANT NUMBER(s) (15) N00014-75-C-0266
9. PERFORMING ORGANIZATION NAME AND ADDRESS Graduate School of Management, University of California, Los Angeles, CA 90024		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR049-345 (12) 24p.
11. CONTROLLING OFFICE NAME AND ADDRESS Information Systems Program, Office of Naval Research, Arlington, Virginia		12. REPORT DATE (11) March 1979
		13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) LEVEL II		15. SECURITY CLASS. (of this report) unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) distribution of this document is unlimited (14) WORKING PAPER-7-79		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Software maintenance; Significant factors in maintenance		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) attached		

407 436 JCB

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PROBLEM FACTORS AND DETERMINANTS IN APPLICATION SOFTWARE MAINTENANCE

Bennet P. Lientz
E. Burton Swanson

March 1979

Information Systems Working Paper #7-79

This work was partially supported by the Information Systems Program, Office of Naval Research, under contract N00014-75-C-0266, project number NR 049-345.

CENTER FOR INFORMATION STUDIES
Graduate School of Management
University of California
Los Angeles, California 90024

Abstract

The problems of application software maintenance in 487 data processing organizations were assessed in a survey. Factor analysis resulted in the identification of six problem factors: user knowledge, programmer effectiveness, product quality, programmer time availability, machine requirements, and system reliability. Of these, user knowledge accounted for about 60% of the total problem variance, providing new evidence of the importance of the user relationship, in accounting for system success or failure. Potential determinants of the problems in maintenance were also analyzed. Problems of programmer effectiveness and product quality were seen to be relatively greater for systems which were older and larger, and where more relative effort in corrective maintenance was spent. All problem factors were positively associated with the level of effort in maintenance. A tendency toward lesser problems in maintenance existed, where the maintenance programmers were also involved in the development of the system. The problem of product quality was seen as lesser, where certain productivity techniques were used in development. Among various organizational controls, only the periodic audit seemed usefully related, by means of its association with lesser problems of user knowledge and product quality. Larger scale data processing environments were significantly associated with greater problems of programmer effectiveness, but with no other problem factor.

Introduction

What are the principal problems in application software maintenance, and how do these problems vary according to the application system maintained and the data processing environment in which maintenance takes place? This paper reports research results which provide some answers to this question.

The results to be reported are based on a sample survey of 2000 managers of data processing organizations. Each manager, randomly selected from the membership of the Data Processing Management Association (DPMA), was mailed a copy of a questionnaire on the subject of application software maintenance. A total of 487 responses were received, a return rate of 24%, considered good given the demanding nature of the questionnaire. Basic descriptive results of the survey are reported in Lientz and Swanson (1979).

The present paper concentrates on the problems of maintenance, as assessed by data processing management, in responding to one item of the questionnaire. Analysis of the responses to this item, considered in conjunction with responses to other questionnaire items, provides insight into both the factors underlying the maintenance problems, and the variables which serve as determinants.

The Problem Items

The questionnaire sought the data processing manager's judgment of the problems in maintaining a selected application system. A list of 26 potential problems was provided, and it was requested that each be evaluated on a 1 to 5 point scale ranging from "no problem at all" to "major problem." The list of potential problems was identical to that used in a preliminary survey (Lientz, et. al., 1978), with the exception of the last three items, which were added on the basis of comments received in this first attempt.

A summary of the descriptive results is presented in Figure 1. The computation of means and variances is based upon the assumption of interval item scales. (For a summary of the same results, based upon ordinal item scaling, see Lientz and Swanson, 1979.) Problem items are listed in the order in which they appeared in the questionnaire. It is seen that "user demands for enhancements and extensions" emerges as the leading problem, a result which validates the findings of the preliminary study (Lientz, et. al., 1978).

The Problem Factors

A factor analysis of the responses to the 26 problem items was performed, in order to explore the underlying dimensionality, and to facilitate further analysis. The principal factor with iteration option was employed, with varimax rotation (Nie, et. al., 1975). A classical factor analysis, based on inferred factors, was thus performed. The method of rotation was the commonly used one.

The factor analysis produced six factors which accounted for the variance in the twenty six problem items. These factors are summarized in Figure 2. The labels attached to the factors are the result of an interpretation of the factor score coefficients shown in Figure 3. These coefficients indicate the relative contributions of the problem items to the factor scores computed. To provide a guide to the interpretation, coefficients of noteworthy size are identified with asterisks (see the key to the Figure). The interpretation proves to be rather straightforward.

It is seen that five of the six factor coefficients marked with an asterisk under Factor 1 identify problem items which refer specifically to the <u>users</u> of the information system. The sixth, which refers to "management

Figure 1

Distributions of Maintenance Problem Items

Variables	Labels	Mean	Standard dev	Cases
MNProb1	Maintenance personnel turnover	2.2332	1.3356	446
MNProb2	Documentation quality	3.0000	1.3103	446
MNProb3	System hardware and software changes	2.0404	1.1739	446
MNProb4	Demand for enhancements and extensions	3.2018	1.1745	446
MNProb5	Skills of maintenance programmers	2.0807	1.1564	446
MNProb6	Quality of original programming	2.5897	1.3256	446
MNProb7	Number of maint. programmers available	2.5762	1.3348	446
MNProb8	Competing demands for programmer time	3.0336	1.3395	446
MNProb9	Lack of user interest	1.8677	1.2171	446
MNProb10	System run failures	1.8677	0.9706	446
MNProb11	Lack of user understanding	2.6076	1.2670	446
MNProb12	Program storage requirements	1.9776	1.2158	446
MNProb13	Program processing time requirements	2.5538	1.2562	446
MNProb14	Maintenance programmer motivation	1.9170	1.1076	446
MNProb15	Forecasting maint. prog. requirements	2.4552	1.2239	446
MNProb16	Maintenance programming productivity	2.0359	1.0866	446
MNProb17	System hardware and software reliability	1.8094	1.0438	446
MNProb18	Data integrity	1.9036	1.0840	446
MNProb19	Unrealistic user expectations	2.5516	1.2616	446
MNProb20	Adherence to programming standards	2.1143	1.0905	446
MNProb21	Management support	1.8453	1.1141	446
MNProb22	Adequacy of system design specs	2.4233	1.2606	446
MNProb23	Budgetary pressures	1.9798	1.2075	446
MNProb24	Meeting scheduled commitments	2.6861	1.2435	446
MNProb25	Inadequate user training	2.7623	1.2387	446
MNProb26	Turnover in user organization	2.3610	1.2351	446

Figure 2

Summary of Maintenance Problem Factors

Factor	Label	Eigenvalue	Pct of Var	Cum Pct
1	User knowledge	7.25414	59.5	59.5
2	Programmer effectiveness	1.45230	11.9	71.4
3	Product quality	1.16047	9.5	80.9
4	Programmer time availability	0.97415	8.0	88.9
5	Machine requirements	0.76567	6.3	95.2
6	System reliability	0.58859	4.8	100.0

Figure 5

Problem Factor Coefficients

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
MNProb1	-0.04521	0.10695*	0.02156	0.00674	-0.05779	0.06764
MNProb2	-0.06644	-0.03852	0.27184**	0.00110	-0.00582	0.00955
MNProb3	-0.00943	-0.01415	-0.00699	-0.01810	0.01991	0.17333*
MNProb4	0.05118	0.02152	-0.01526	0.02296	0.03172	-0.02123
MNProb5	-0.08622	0.22730**	0.03769	-0.03784	-0.02746	0.04512
MNProb6	-0.08716	-0.01115	0.32069**	-0.04889	-0.01086	0.01871
MNProb7	-0.03169	0.07115	-0.00522	0.14772*	-0.01257	-0.02537
MNProb8	-0.06997	-0.17608*	-0.01721	0.78537**	-0.02472	-0.00635
MNProb9	0.13855*	-0.01215	0.02271	-0.05349	-0.01204	-0.09485
MNProb10	-0.02340	-0.02459	0.10837*	-0.02321	-0.00812	0.15754*
MNProb11	0.36301**	-0.10578*	-0.02331	-0.07230	0.01735	-0.09868
MNProb12	-0.03805	-0.01484	-0.01234	-0.02045	0.47591**	-0.03657
MNProb13	-0.03105	-0.02690	-0.01227	-0.01694	0.47141**	0.06795
MNProb14	-0.01147	0.34931**	-0.04861	-0.09634	0.00968	-0.10137*
MNProb15	0.03589	0.14128*	-0.10887*	0.10180*	-0.00026	-0.04935
MNProb16	-0.02935	0.36939**	-0.10989*	-0.08001	0.03882	-0.02649
MNProb17	-0.07293	-0.01827	-0.00941	-0.02913	-0.02346	0.43967**
MNProb18	0.05905	-0.01681	0.03245	-0.03812	-0.08623	0.22349**
MNProb19	0.17311*	-0.04776	-0.02467	0.03369	-0.04264	0.00607
MNProb20	-0.01215	0.06375	0.07826	-0.00841	0.01669	0.02722
MNProb21	0.15589*	-0.00798	-0.00157	-0.02836	0.02734	-0.05526
MNProb22	0.03377	-0.07672	0.40420**	-0.02800	-0.02223	-0.16507*
MNProb23	0.07904	-0.00187	-0.02341	0.01216	0.00309	0.00032
MNProb24	0.05503	0.03677	0.03999	0.07347	0.02871	-0.05145
MNProb25	0.23669**	-0.04170	-0.12469*	0.00565	-0.04806	0.11472*
MNProb26	0.12929*	0.00949	-0.12001*	0.00232	-0.05121	0.13258*

Asterisk key: ** indicates absolute value of coefficient equal to or greater than .200
* indicates absolute value of coefficient equal to or greater than .100, but less than .200

support," may also be given a user-related interpretation. Factor 1, which accounts for 59.5% of the variance in the responses to all 26 problem items, may thus be reasonably labeled the problem of user knowledge, based in particular upon the two largest factor coefficients, which refer to "lack of user understanding" and "inadequate user training." The labeling of the five other factors follows similar lines of reasoning, and provides no major difficulties in interpretation.

On the whole, the results of the factor analyses are striking. The dimensions emerge with unexpected clarity, and the dominance of the oft-cited "user problem" is remarkable. It is not simply that user problems are common to all; it is that user problems account for the major variance in the problems common to all. Note in particular that problem item 4, "user demand for enhancements and extensions," while it is the major problem item among all those cited, does not appear as a significant component in Factor 1.

For purposes of subsequent analysis, the six problem factors were formalized as indices, computed on the basis of problem items with factor coefficients of absolute value 0.200 or greater. Normalized values of the problem item scores were used in the indices (Nie, et. al., 1975). A summary of the six factor indices and their problem item components is presented in Figure 4.

Problem Determinants

What are the determinants of the problems of application software maintenance? To address this question, relationships between the problem factors and other maintenance variables were investigated.

(i) Application system size and age

Five measures of application system size were obtained in the survey: number of programs, number of source statements, number of files, number of

Figure 4

Problem Factor Indices and their

Item Components

Factor Index	Item Component
1. User knowledge	11. Lack of user understanding (.363)
	25. Inadequate user training (.237)
2. Programmer Effectiveness	16. Maintenance programming productivity (.369)
	14. Maintenance programming motivation (.349)
	5. Skills of maintenance programmers (.227)
3. Product Quality	22. Adequacy of system design specs (.404)
	6. Quality of original programming (.321)
	2. Documentation quality (.272)
4. Programmer Time Availability	8. Competing demands for programmer time (.785)
5. Machine Requirements	12. Program storage requirements (.476)
	13. Program processing time requirements (.471)
6. System Reliability	17. System hardware and software reliability (.440)
	18. Data integrity (.223)

Note: Factor score coefficients shown in parentheses

data base bytes, and number of pre-defined user reports. Where parametric analyses were performed, natural logarithm transformations of these measures were judged necessary, to meet normality assumption requirements.

Larger systems proved to be significantly associated with greater problems in maintenance. Of 30 first-order Pearson correlation coefficients computed between the six problem factors and the five measures of system size, 26 were positive, of which 22 were significant at the $s \leq 0.100$ level, and 14 were of magnitude $r \geq 0.100$. The problem factor "programmer effectiveness" demonstrated a notable positive association with all five measures of system size. The factor "product quality" was positively associated with four of the five size measures.

The age of the application system, measured in terms of the number of months since the system became operational, was also obtained from the questionnaire. As with system size, first-order Pearson correlation coefficients between the six problem factors and system age were computed.

Older systems tended to be perceived as having greater problems in maintenance. In particular, system age was positively and notably associated with problems of product quality ($r = 0.142$, $s = 0.001$) and programmer effectiveness ($r = 0.128$, $s = 0.003$). The other correlations were not notable, however.

Though system size and age are seen to be strongly associated with the problems of maintenance, this association was shown in subsequent analysis to be explainable in terms of other, intervening variables.

(ii) Magnitude and allocation of the maintenance effort

The magnitude and allocation of the maintenance effort on the application system described were also included among the data obtained. Two measures of the magnitude of effort were obtained: the total number of individuals assigned (in whole or in part) to maintenance of the system, and the total

number of person-hours expended annually. The allocation of the maintenance effort was indicated by a percentage breakdown of annual person-hours according to eight categories: (i) emergency program fixes; (ii) routine debugging; (iii) accommodation of changes to data inputs and files; (iv) accommodation of changes to hardware and system software; (v) enhancements for users; (vi) improvement of program documentation; (vii) recoding for efficiency in computation; and (viii) others. The categories chosen were based on the classification system originally proposed by Swanson (1976). Within this system, emergency program fixes and routine debugging comprise <u>corrective maintenance</u>; accomodations of change represent <u>adaptive maintenance</u>; and user enhancements, improved documentation, and recoding for efficiency make up <u>perfective maintenance</u>.

As with the measures of application system size, natural logarithm transformations of the two measures of the magnitude of the maintenance effort were judged necessary, for parametric analysis purposes.

The problems of maintenance were perceived to be the greater, the greater the magnitude of the effort in maintenance. The first-order Pearson correlation coefficients are shown in Figure 5. All twelve coefficients are positive, eleven of these are statistically significant at the $s \leq 0.100$ level, and eight are of notable magnitude $r \geq 0.100$. The correlations between number of maintenance person-hours and the problems of programmer effectiveness ($r = 0.263$) and product quality ($r = 0.240$) are particularly striking.

Problems of maintenance were also perceived to be the greater, the more relative time is spent in corrective maintenance. Of twelve first-order Pearson correlation coefficients relating the six problem factors to relative time spent in emergency fixes and routine debugging, eleven were positive, of which nine were statistically significant and five were notable. Relative time in emergency fixes was positively associated with problems of product

Figure 5

Maintenance Problem Factors and Magnitude of Maintenance Effort: First-order Pearson Correlation Coefficients

	Annual Person-hours	Persons Assigned
PFactor1 (User knowledge)	0.1158* (451) P=0.007	0.0965 (461) P=0.019
PFactor2 (Programmer effectiveness)	0.2625** (447) P=0.000	0.2088** (456) P=0.000
PFactor3 (Product quality)	0.2404** (499) P=0.000	0.1099* (459) P=0.009
PFactor4 (Programmer time availability)	0.1445* (453) P=0.001	0.0879 (462) P=0.029
PFactor5 (Machine requirements)	0.0949 (452) P=0.022	0.0502 (462) P=0.141
PFactor6 (System reliability)	0.1372* (452) P=0.002	0.1186* (461) P=0.005

Asterisk key: ** indicates $|r| \geq 0.200$

* indicates $0.100 \leq |r| < 0.200$

quality ($r = 0.200$, $s = 0.001$); user knowledge ($r = 0.130$, $s = 0.002$); and programmer effectiveness ($r = 0.117$, $s = 0.005$). Relative time in routine debugging was positively associated with problems of product quality ($r = 0.204$, $s = 0.001$) and programmer effectiveness ($r = 0.132$, $s = 0.002$).

Two other relationships involving the allocation of the maintenance effort with problem factors were of notable significance and magnitude. The problem of machine requirements was positively associated with both recoding for computational efficiency ($r = 0.164$, $s = 0.001$) and accommodating system hardware and software changes ($r = 0.106$, $s = 0.010$). Both these relationships are easily understood.

Interestingly, no notable findings related the percent time spent in providing user enhancements to any of the problems of maintenance, including that of user knowledge.

(iii) Relative development experience of maintainers of the system

It was asked in the questionnaire how many of the individuals currently assigned to the maintenance of the application system had worked previously on the development of this same system. The number who had, divided by the total, thus served a measure of the relative development experience of the maintainers, with respect to the system being maintained.

The computation of first-order Pearson correlation coefficients showed relative development experience to be significantly related to perceived problems in maintaining the application system. The most significant relationships indicate greater development experience to be associated with lesser problems with product quality ($r = -0.270$, $s = 0.001$) and lesser problems with programmer effectiveness ($r = -0.171$, $s = 0.001$). Lesser problems with user knowledge and programmer time availability were also indicated for higher levels of relative development experience, but correlation coefficients were not of magnitude $r \geq 0.100$. Greater problems with

machine requirements were indicated (r = 0.104, s = 0.011), for which there is no obvious interpretation. No relationship to the problem of system reliability existed.

(iv) Use of productivity techniques in system development

It was asked in the questionnaire which of a variety of tools, methods and techniques were employed in the development of the application system described. Included in the checklist were: decision tables, data base dictionary, test data generators, structured programming, automated flow-charging, HIPO (Hierarchy plus Input-Process-Output), structured walk-through, and chief programmer team. Provision for "others" to be indicated was also included.

One-way analyses of variance of the six problem factors according to the use of the productivity tools were performed. The results showed the problem of product quality to vary significantly according to use of five of the eight tools listed. Specifically, the software product is perceived to be of better quality (in terms of the three components from which the factor is derived: system design specifications, programming, and documentation) where test data generators, structured programming, HIPO, structured walk-through, or the chief programming team have been employed. These results should be heartening to advocates of these techniques. However, it is also noteworthy that the problems of user knowledge and programmer effectiveness, which account together for 71.4% of the total problem variance, are little affected through the use of these same techniques.

(v) Use of a data base management system

Whether the application system made use of a data base management system was also reported in the questionnaire. Analyses of variance of the six problem factors according to the use of a data base management system were thus performed. No significant variances were found, and it may be concluded that management's assessment of the problems in maintenance is likely to be the same, on average, for application systems supported by a

DBMS, as for application systems unsupported.

(vi) Programming language

Also examined were the variances in the six maintenance problem factors according to the principal language in which the application system was programmed.

Employing one-way analyses of variance, two problem factors were seen to vary significantly according to programming language used: programmer effectiveness ($s = 0.010$) and system reliability ($s= 0.080$). In the case of programmer effectiveness, problems tend to be slightly greater than average where assembler languages are used, and notably greater than average where FORTRAN or PL/1 is used. Where COBOL is used, problems of programmer effectiveness are about average, and where RPG is used, notably less than average. On the whole, it first appears, the more "sophisticated" the language, the greater are the problems in programmer effectiveness.

However, the interpretation of a direct causal relationship between programming language and the problem of programmer effectiveness proves not to be warranted. Other analysis suggests that the scale of the data processing environment may account for a portion of the relationship, since more sophisticated languages tend to be used in the larger environments, and the problems of programmer effectiveness tend also to be greater, the larger the organization. (See the discussion under "Relationships among determinants.")

In the case of system reliability, one-way analyses of variance showed the problem to be greater than average where Fortran is used, and less than average where RPG is used. For the other languages the problem is about average. Since other analysis indicates the problem of system reliability does not vary significantly with the scale of the data processing environment, the present apparent relationship may not be explained as with programmer effectiveness. Any interpretation here is tenuous, but it may be that RPG

software is seen as more reliable, on the average, for application programming, and FORTRAN is seen as less reliable.

(vii) Use of organizational controls

Also included in the questionnaire was a checklist of organizational controls which might be established for the maintenance of the application system. Listed were: (i) logging and documentation of user requests; (ii) cost justification of user requests; (iii) logging and documentation of troubles in operational processing; (iv) logging and documentation of changes to programs; (v) formal retest procedure for program changes; (vi) batching of program changes according to a predetermined schedule; (vii) periodic formal audit; (viii) equipment cost charge-back system; (ix) personnel cost charge-back system. Respondents were asked to indicate which of the above controls were used in the maintenance of the application system described.

One-way analyses of variance of the six problem factors according to the use of organizational controls were first performed. A total of 16 relationships were established at significance levels ranging from $s < 0.001$ to $s = 0.070$. Of these, 13 relationships were positive, suggesting that, on the whole, the use of organizational controls is associated with relatively greater problems in maintenance.

However, further analysis showed these results to be explainable in terms of other relationships. When these relationships are taken into consideration, the use of organizational controls proves to be little related to the problems of maintenance. (This is discussed further under "Relationships among determinants.") An exception is the periodic audit, which emerges as significantly associated with lesser problems of user knowledge and product quality.

(viii) The data processing environment

The first part of the questionnaire sought to establish the data processing environment in which maintenance of the application system described took place. The industry served by the data processing department was identified. The scale of the department was specified, in terms of both personnel and annual equipment budget, and the overall organization and allocation of staff time between maintenance and new system development activities was indicated. The relative demands of maintenance on the data processing manager's own time was assessed, as was the current level of departmental staffing.

The scale of the data processing department, as measured by the annual equipment budget, proved to be strongly related to the perceived problems in maintaining the application system described. Specifically, analysis of variance showed the problem of programmer effectiveness to be perceived as greater, the larger the scale of the department ($s < 0.001$, linearity also significant at the $s < 0.001$ level). No significant variances in the other five problem factors were found. Thus, the perceived problems of maintenance vary by the size of the organization along the single dimension of "programmer effectiveness." Two interpretations are suggested for consideration. The straightforward interpretation is that smaller organizations do have greater programmer effectiveness, possibly because the advantages of simplicity of work coordination outweigh the disadvantages of lack of technical specialization. An alternative interpretation is that programmer effectiveness does not itself vary, but that awareness of the "problem" is heightened according to the visibility of the data processing budget. Other interpretations may also be possible.

Maintenance problem factors were also correlated to the percent time spent on maintenance in the organization as a whole. As might be expected, problems are seen to be the more severe, the more of the organization's time

is allocated to maintenance. Four of the six problem factors, accounting together for 88.9% of the problem variance, are of notable significance and magnitude: programmer effectiveness ($r = 0.191$, $s = 0.001$); product quality ($r = 0.158$, $s = 0.001$); user knowledge ($r = 0.113$, $s = 0.006$); and programmer time availability ($r = 0.101$, $s = 0.013$). System reliability is also positively related, though the magnitude is not notable. The problem of machine requirements is unrelated.

Finally, analyses of variance of the six problem factors by the perceived level of staffing sufficiency and the demands of maintenance on the manager's own time were performed. It was found that four problem factors varied significantly by perceived level of staffing sufficiency: user knowledge ($s = 0.007$); programmer effectiveness ($s < 0.001$); product quality ($s < 0.001$); and programmer time availability ($s < 0.001$). In each case, linearity was also significant at the $s = 0.003$ level or better, and it may be concluded that each problem tends to be perceived as greater, as the staffing level is regarded as less sufficient, on the whole. These are strong results, though not particularly surprising.

In the case of the demands of maintenance on the manager's own time, four problem factors also varied significantly: user knowledge ($s = 0.005$); programmer effectiveness ($s = 0.055$); product quality ($s < 0.001$); and system reliability ($s = 0.076$). Again, linearity was also significant in each case, here at the $s = 0.016$ level or better, and it may be concluded that each problem tends to be perceived as greater, the more of the manager's own time is absorbed by the demands of maintenance. As before, these results are what might be expected.

Relationships Among Determinants

Various of the determinants discussed in the previous section were

themselves interrelated. These relationships proved to explain certain of the initial findings.

A principal network of relationships involved system age and size, magnitude of the maintenance effort, relative allocation of effort to corrective maintenance, and the relative development experience of the maintainers. Preliminary analysis suggested that the impact of these variables upon the problems of maintenance should be considered jointly. Based upon a causal structure suggested by the preliminary analysis, a series of multiple regression analyses were performed, with the independent variables introduced hierarchally. Two measures of the magnitude of the maintenance effort were introduced first, followed by the two component measures of corrective maintenance, the five measures of system size, the single measure of relative development experience, and, lastly, system age. At each level of the hierarchy, measures were introduced according to their statistical significance at that step. (See Nie, et. al., 1975, for a discussion of the analytical procedure.)

Results confirm that the magnitude of the effort in maintenance, the allocation of this effort to corrective maintenance, and relative development experience are all of importance in explaining the problems of maintenance. In particular, number of maintenance person-hours accounts for a substantial portion of the problem of programmer effectiveness, and the relative development experience of the maintainers and the percent time on corrective maintenance (which includes both emergency fixes and routine debugging) are of similar significance in accounting for the problem of product quality.

However, results also indicate that system size and age have little influence upon the problems of maintenance, apart from their established impact upon the magnitude and allocation of the maintenance effort, and the relative development experience of the maintainers. When the latter

variables have been entered first into the regression equations, measures of system size have notable influence only upon the problems of machine requirements and system reliability, which together account for only 11.1% of the total problem variance. System age has no notable influence whatsoever.

A similar analysis was performed with respect to the impact of the use of organizational controls upon the perceived problems in maintenance. As discussed earlier, one-way analyses of variance suggested that, on the whole, the use of organizational controls is associated with relatively greater problems in maintenance. However, multiple regression analysis showed this association to be explainable in terms of relationships already established. A series of analyses were performed, with hierarchic entry of the independent variables. Entered first were the number of annual maintenance person-hours, the relative allocation of effort to emergency fixes and routine debugging, and the relative development experience of the maintainers. Organizational controls were then permitted to enter the equations. Results indicated that few associations between the problems of maintenance and the use of organizational controls exist, when the control variables have been accounted for.

Finally, a relationship between the scale of the data processing environment and the choice of programming language proved to be of importance in identifying one apparently spurious relationship among the initial findings. As discussed earlier, initial findings suggested that more sophisticated programming languages were associated with greater problems of programmer effectiveness. However, it was also found that more sophisticated languages tend to be used in the larger scale environments. A two-way analysis of variance of the problem of programmer effectiveness was thus conducted, controlling for the size of the data processing equipment budget, in addition to the programming language used. (A hierarchical approach to the partitioning

of main effects was employed, assigning priority to size of the data processing equipment budget, assumed to have causal precedence. See Nie, et. al., 1975.) Results indicated that the main effects of programming language are not significant at the s ≤ 0.100 level, when the size of the equipment budget is controlled. Thus, the apparent relationship between programming language and the problem of programmer effectiveness is explained in terms of the data processing environment in which programming takes place. It may be that larger installations are characterized by more complex applications, for which more sophisticated languages are suited, and which, at the same time, present greater challenges to effective programming. However, other explanations are also possible.

Conclusion

The problem factor, user knowledge, has been seen to account for the majority of the variance in data processing management's assessment of the problems in maintenance. This result provides further evidence of the importance of the relationship between the users and the providers of information systems, in the determination of system success or failure.

"Lack of user understanding" and "inadequate user training" are the two components of the problem factor, user knowledge, and both suggest an estrangement of the users from the systems intended to serve them. It is this aspect of the oft-cited "user problem" which distinguishes systems with relatively lesser maintenance problems from those with maintenance problems relatively greater.

"User demands for enhancements and extensions" is the problem item seen as most severe, overall. This item is not a component of the user knowledge factor, though it may be somewhat related. It does not account for variance in the problems of maintenance. Rather, it is the common

complaint. That it is such is perhaps understandable, when it is recalled that user demand is not controllable by data processing management, but yet has substantial implications for its resource allocation decisions.

The potential determinants investigated proved most strongly related to problems of programmer effectiveness and product quality. When the nature of these determinants is considered, this result is, in part, understandable. The determinants consisted primarily of characteristics of the software, the programming staff, the programming effort, and the programming environment. Characteristics of the user environment were not included. Such characteristics would presumably be more strongly related to the problem of user knowledge.

The direction for future research thus seems clearly indicated. Attention should be focused on the users of data processing applications. Characteristics of user environments in which relatively successful applications exist should be identified. These characteristics should then be related to their impact upon the maintenance of the application software, as performed by the data processing department.

A likely approach would be to study the problems of ongoing systems from the viewpoints of users, as opposed to data processing professionals. Problem factors, from the users' perspective, might be identified and correlated to the corresponding problem factors, as seen by data processing management. An interesting question is whether the user knowledge problem factor would be recognized by the users themselves. If not, how would the users see the problem? Further, what characteristics of user environments would explain the variance in their problem perceptions?

By approaching the "user problem" from both sides, a more enlightened understanding of the problems of application software maintenance should ultimately be expected.

References

1. Lientz, B.P., E.B. Swanson, and G.E. Tompkins. "Characteristics of Application Software Maintenance," Communications of the ACM, Vol. 21, No. 6, June 1978, pp. 466-471.

2. Lientz, B.P. and E.B. Swanson. "Software Maintenance: A User/ Management Tug-of-War," Data Management, Vo. 17, No. 4, April 1979, pp. 26-30.

3. Nie, Norman H., C.H. Hull, J.G. Jenkins, K. Steinbrenner, and D.H. Bent. SPSS: Statistical Package for the Social Sciences. 2nd Edition, McGraw-Hill, 1975.

4. Swanson, E.B. "The Dimensions of Maintenance," Proceedings of the Second International Conference on Software Engineering. San Francisco, October 13-15, 1976.